

**REMARKS**

Claims 2-46, 48-78 and 80-87 are pending for consideration. In view of the following remarks, Applicant respectfully requests that this application be allowed and forwarded on to issuance.

**The § 102 Rejections**

Claims 1-22, 24-32, 34-38 and 40-87 stand rejected under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent No. 6,460,058 to Koppolu et al. (hereinafter "Koppolu").

**The § 103 Rejections**

Claims 23, 33 and 39 stand rejected under § 103(a) as being unpatentable by Koppolu in view of U.S. Patent No. 6,606,606 to Starr (hereinafter "Starr").

Before discussing the substance of the Office's rejections, the following discussion of Applicant's disclosure as well as the Koppolu reference is provided in an attempt to assist the Office in appreciating the patentable distinctions between Applicant's claimed subject matter and the cited reference.

**Applicant's Disclosure**

Perhaps a good place to start in describing the material of Applicant's disclosure is with the environment in which Applicant's embodiments operate. Specifically, the current popular paradigm in software application design is to provide a separate and oftentimes

1 different window for each application that might be executing on a  
2 computing device. When a user desires to use a particular application,  
3 they will typically open up the application which will then be presented by  
4 the computer system in the form of a window or windows that allows the  
5 user to interact with the application. For example, a word processing  
6 application (such as Microsoft Word) will typically be displayed in a  
7 window that has an area for a user to create a document or retrieve and edit  
8 a document that is already in existence. If a user wishes to read their  
9 email, then they will typically open up an email application (such as  
10 Microsoft Outlook) which will typically be displayed in another window  
11 that is separate and different from the word processing window. Now, the  
12 user has two windows to manage—a word processing window and an  
13 email window.

14 Traditionally, a user has managed multiple windows through the use  
15 of a task bar that is, many times, located at the bottom portion of a user  
16 display. The task bar is a very thin bar that can extend across a good  
17 portion of the user's display and includes references to applications or  
18 documents that the user is using or has used. The user can "minimize" the  
19 email window when, for example, they wish to work inside the word  
20 processing application. When the email window is minimized, a reference  
21 to the email application is placed in the task bar. If the user receives an  
22 email during the course of working in the word processing application,  
23 they can restore the email window by clicking on the reference to the  
24 email application in their task bar. This restores the email application so  
25 that the user can interact with and read their email. In the course of

1 reading an email message, the message is likely to be presented in yet  
2 another window. If the user chooses to respond to the email message, they  
3 typically prepare their response in a "reply" window which is yet a fourth  
4 window that the user must manage.

5 The above-described scenario constitutes a simple example of a  
6 window management scenario when the user has opened only two  
7 applications. Consider the case where a user has multiple applications  
8 (e.g. four or more) that they are working in throughout the day. As a  
9 specific example, consider the following four exemplary applications that  
10 a user might find necessary to use during the course of their computing  
11 day: a word processing application, a presentation application (such as  
12 Microsoft Powerpoint), a web browser application (such as Microsoft  
13 Internet Explorer), and an email application.

14 What many users typically do is they open all of the applications  
15 and then manage each application's separate and different window as they  
16 need access to a different application. The application managing part of  
17 the screen uses what is referred to as a "windowing environment." Yet,  
18 windowing environments are not necessarily intuitive to all potential  
19 computer users. User studies have consistently shown that one of the  
20 biggest hurdles for new users of a windowing environment is learning to  
21 understand a windowing system, where windows can "layer" on top of  
22 each other. Consider, for example, Fig. 1 which shows an exemplary user  
23 display 10 that includes four exemplary windows 12, 14, 16, 18,  
24 respectively dedicated to a word processor application, a presentation  
25 application, a web browser application, and an email application. These

1 windows are layered on top of one another which, for a new computer  
2 user, can be a difficult concept to understand and manage. For example,  
3 some users might not understand how to use a task bar to manage the  
4 windows. They might, for example, inadvertently close an application  
5 when they simply intended to minimize it. Additionally, the users might  
6 not appreciate or understand how to move the separate windows around on  
7 their display. Further, different windows can sometimes be inadvertently  
8 clicked by a user. For example, while a user is in a window associated  
9 with one application, they may inadvertently click the edge of a window  
10 associated with another application, whereupon they find themselves in the  
11 middle of a different application. Finally, many users simply do not  
12 comprehend that their screen has multiple layers: they only think of the top  
13 window as the one that they can interact with, as if the previous ones were  
14 lost.

15       Windowing environments, however, do not just pose challenges to  
16 newer computer users: they can sometimes pose challenges to users who  
17 are familiar with such environments. Specifically, management of  
18 multiple windows can be distracting to computer users, particularly when a  
19 user has many different windows that they are attempting to manage. For  
20 example, if a user has many different applications that they are executing  
21 that are being managed by a task bar at the bottom of the user's display, to  
22 switch from one application to another, the user must find the appropriate  
23 task bar portion that references an application of interest. The user must  
24 then click on the task bar portion to pull up the application. This scenario  
25 can be complicated if, for example, the user has multiple documents in one

1 application that have been minimized, e.g. multiple word processing  
2 documents or multiple email messages that they might intend to respond to  
3 during the course of their day. This complicates the scenario because now  
4 the task bar must maintain an entry for not only each of the user's  
5 applications, but each of the documents within each application that might  
6 have been minimized by the user as well. At this point, accessing the  
7 minimized applications or document is not an easy task, but rather has  
8 devolved into a trial and error hunting process. This is not an efficient way  
9 for busy users to manage their applications and documents.

10 Consider now the approach described in Applicant's disclosure for  
11 addressing these problems. Specifically, in accordance with one  
12 embodiment, software provides a user interface (UI) that presents a user  
13 with a single navigable window that can be navigated from functionality to  
14 functionality and inside individual functionalities by a user. The user  
15 interface enables the user to effectively manage multiple windows, and  
16 hence multiple functionalities, by presenting only one window at a time.  
17 This is different from the traditional windowing environment because the  
18 windows that pertain to the individual functionalities are not layered on  
19 one another and do not need separate management. *Another noteworthy*  
20 *way that the single navigable window varies from the traditional*  
21 *windowing environment is that the various functionalities are provided*  
22 *by a single application.* That is, in the traditional windowing  
23 environment, it is very typical for multiple windows to be provided by  
24 multiple different applications that are opened by a user, e.g. a word  
25 processing application will have one window, an email application will

1 have another window, a web browsing application will have another  
2 window, and the like. All of these windows are separate and requirement  
3 separate management by the user. *In the present case, various*  
4 *functionalities that were once the domain of individual separate*  
5 *applications are now the domain of a single integrated application which*  
6 *provides its own window management scheme.* The window management  
7 scheme is embodied in the form of a single navigable window that can be  
8 navigated by a user from functionality to functionality.

9 A user, through the use of navigation instrumentalities can navigate  
10 between the functionalities and when doing so, the single window ensures  
11 that only one of these functionalities is presented to a user at a time. In  
12 one embodiment, one navigation instrumentality is provided in the form of  
13 a web browser-like navigation tool. The choice of a web browser-like  
14 navigation tool follows from concerns that navigation instrumentalities be  
15 of a type that is readily understood by most individuals familiar with  
16 computing environments. Thus, when a user first encounters the inventive  
17 navigable single window concept for the first time, they do not have to  
18 learn an unfamiliar navigation concept. Another navigation instrumentality  
19 includes, in accordance with another embodiment, links to each of the  
20 multiple different functionalities. These links can be clicked on by a user  
21 and the single navigable window is automatically navigated to the selected  
22 functionality.

23 Fig. 3 shows but one exemplary user interface (UI) 300 in  
24 accordance with one described embodiment. In the illustrated example, UI  
25

1 300 includes a navigation bar 302, one or more command areas 304, and a  
2 display or document area 306 that constitutes the single navigable window.

3 Navigation bar 302 is located adjacent the top of display area 306  
4 and contains browser-like navigation buttons 308 in the form of a  
5 "backward" button, a "forward" button, a "stop" button and the like. The  
6 navigation bar can be located anywhere on the UI. Its illustrated  
7 placement, however, is similar in appearance to the placement of  
8 traditional web browsing navigation features. In addition to the navigation  
9 buttons 308, the navigation bar 302 also includes, in accordance with one  
10 embodiment, links 310 to the different functionalities that can be accessed  
11 by the user. In the illustrated example, links to three exemplary  
12 functionalities (i.e. functionality 1, functionality 2, and functionality 3) are  
13 shown. These functionalities are typically different functionalities that can  
14 enable a user to complete different respective tasks. Examples of different  
15 tasks are given below in more detail. These functionalities are all provided  
16 within the context of a *single application*. To access a particular  
17 functionality, a user simply clicks on one of the links and a window that  
18 pertains to the selected functionality is immediately presented in the  
19 display area 306.

20 In accordance with one embodiment, command areas 304 are  
21 located adjacent the top and left side of the display area 306. The  
22 command area(s) can, however, be located in any suitable location. The  
23 command areas provide commands that are both global in nature and  
24 specific to the particular context the user has selected. For example, some  
25 commands such as "search" and "help" might be considered as global in

1 nature since they can find use in many contexts. Other commands, such as  
2 "text bold" or "forward" are more specific to the particular context that the  
3 user has selected. For the "text bold" command, the user's context may  
4 likely be a word processing context, while the "forward" command may  
5 likely be employed in an email context.

6 As an example of the single navigable window provided by a *single*  
7 *application* consider Figs. 4 and 5.

8 In this example, the multiple functionalities 310 that can be  
9 navigated by a user include a browser functionality (indicated by the home  
10 icon), a mail functionality (indicated by the letter icon), a planner  
11 functionality (indicated by the clock icon), a contacts functionality  
12 (indicated by the people icon), a documents functionality (indicated by the  
13 folder icon), and a links functionality (indicated by the world icon). These  
14 functionalities are so-called "document-centric" functionalities because  
15 they all relate in some way to a document that a user interacts with, e.g. a  
16 Web page document, an email document, a calendar document, etc.

17 Fig. 4 shows an example of a display that is rendered in the display  
18 area 306 when a user clicks on the link to the browser functionality. By  
19 clicking on the link (i.e. the home icon) to the browser functionality, *single*  
20 *application program software* executing on the user's computer executes  
21 to implement a browser functionality. In this example, the browser  
22 functionality displays the user's home page in display area 306. Notice  
23 also that navigation buttons 308 are provided for navigation between the  
24 different selectable functionalities. The command areas 304 contain  
25 command sets that include commands that are specific to the context that



1 the user has selected. In this example, the user's context is a browsing  
2 context. Accordingly, the leftmost command area contains commands that  
3 are specific to the browsing functionality. Such commands include ones  
4 that a user would normally expect to find in a web browser. Notice also  
5 that the command area 304 adjacent the top of display area 306 also  
6 contains commands that are specific to the browsing context, i.e. "Add to  
7 Favorites" and an address well in which the user can type a URL of a  
8 particular destination web site.

9 Fig. 5 shows an example of a display that is rendered in the display  
10 area 306 when the user clicks on the link to the mail functionality (i.e. the  
11 folder icon). By clicking on this link, *single application program*  
12 *software* executing on the user's computer executes to implement the mail  
13 functionality. In this example, the mail functionality displays a user's in  
14 box with messages that have been received by the user. Notice that the  
15 leftmost command area has been minimized by the user and that the  
16 command area adjacent the top of the display area 306 contains commands  
17 that are specific to the user's current context, e.g. "New" for generating a  
18 new email message, "Reply" for replying to an email message, "Reply to  
19 All" for replying to all recipients of an email message and the like.

20 Likewise, although not specifically illustrated, the user could have  
21 displays for the planner, contacts, documents, and links functionalities  
22 presented in the display area 306 by simply clicking on the links to these  
23 specific functionalities. The navigation bar 308 provides the user with the  
24 ability to navigate through these different functionalities in a browser-like  
25 manner.

### The Koppolu Reference

Koppolu describes a unified browsing environment based on a hyperlink navigation browsing model. Koppolu describes an object-oriented framework and system having objects, programming interfaces, object integration interfaces, and methods for supporting unified navigation among different varieties of documents, data, and *application programs*. Koppolu instructs that its disclosure overcomes the limitation in previous Internet browsers of navigating within a single application window, and provides seamless navigation between multiple document types and applications. See, e.g. column 3, lines 20-30.

Koppolu instructs that its object-oriented framework includes a hyperlink object which encapsulates information needed to navigate a hyperlink, and the action of navigating the hyperlink. The hyperlink object is described as a system-defined object which supports an integration interface that allows *other programs* to use the hyperlink object for navigating the hyperlink. The hyperlink object contains a moniker to a target of the hyperlink, a character string designating a destination location within the target, a descriptive name for the hyperlink ("friendly name"), and other parameters. See, e.g. column 3, lines 30-42.

Koppolu's object-oriented framework is described to further include a browse context object and integration interfaces for a hyperlink container object, a hyperlink target object and a hyperlink frame object. The browse context object maintains a history of navigated hyperlinks, and other context information for hyperlink navigation including window size

1 and position. *The integration interfaces of these objects allow multiple*  
2 *different applications to interact with the browse context so as to*  
3 *participate fully in hyperlink navigation.* With the browse context and  
4 integration interfaces, *the applications* can provide "go back" and "go  
5 forward" controls which retrace prior hyperlink navigation activity  
6 between *the applications*. *The applications* also can use the browse  
7 context to position and size the destination application window of the  
8 hyperlink so as to *create the appearance of replacing the originating*  
9 *application window of the hyperlink*. The system thereby allows seamless  
10 hyperlink navigation between *different document types and applications*.  
11 See, e.g. column 3, lines 42-61.

12 Koppolu's system also includes a set of programming interfaces  
13 which allow *applications* to participate in unified browsing without  
14 implementing objects to support the integration interfaces. See, e.g.  
15 column 3, lines 61-65.

### 16 17 The Claims

18 **Claim 4** recites a computing system comprising [emphasis added]:

- 19
- 20 • *a single application program* configured to provide:
    - 21 o a single navigable window;
    - 22 o multiple different functionalities to which the single
    - 23 navigable window can be navigated by a user;
    - 24
    - 25

- o a navigation model that is configured to manage the user's navigation activities within *the single application program*; and
- o navigation instrumentalities comprising browser-like navigation buttons associated with the single navigable window, the navigation instrumentalities being configured for use by the user to navigate the single window inside individual functionalities and to the different functionalities.

In making out the rejection of claim 4, the Office argues that Koppolu anticipates this claim. Specifically, the Office argues that Koppolu discloses a single application program configured to provide a single navigable window and cites to Koppolu's Fig. 3 and 5 in support therefore. Additionally, the Office argues that Koppolu discloses multiple different functionalities to which the single navigable window can be navigated citing to column 8, lines 29-40 in support therefore. Further, the Office argues that Koppolu discloses a navigation model as recited in the claim citing to column 10, lines 1-29. Further, the Office argues that Koppolu discloses providing navigation instrumentalities associated with the single navigable window for use to navigate the window inside individual functionalities and to the different functionalities. For the reasons set forth below, Applicant respectfully traverses the Office's rejections.

As noted above, this claim recites a *single* application program configured to provide the single navigable window and the multiple different functionalities. Koppolu's column 8, lines 29-40, which the Office cites for the proposition of a single application program that

1 provides the single navigable window which can be navigated to the  
2 multiple different functionalities states as follows:

3  
4 The browser can be integrated with the operating system  
5 software, or can be a separate application software. The illustrated  
6 browser is implemented by the software as an OLE object  
7 (described below).

8 The illustrated remote computer network 52 is the Internet,  
9 which is described in the Background and Summary of the  
10 Invention above. In the illustrated browsing environment 50, the  
11 computer 20 connects to the computer network 52 over a telephone  
12 line 54 with a modem 56. Other physical connections to the  
13 computer network alternatively can be used, such as an ISDN, T1  
14 or like high speed telephone line and modem, a television cable and  
15 modem, a satellite link, an optical fiber link, an Ethernet or other  
16 local area network technology wire and adapter card, radio or  
17 optical transmission devices, etc.

18 This excerpt in no way describes or even suggests *a single*  
19 *application program* configured to provide a single navigable window and  
20 *multiple different functionalities* to which the single navigable window  
21 can be navigated by a user.

22 Furthermore, Koppolu discloses that the applications "can use the  
23 browse context to position and size the destination application window of  
24 the hyperlink so as to *create the appearance* of replacing the originating  
25 application window of the hyperlink" (Column 3, Lines 56 – 61)(emphasis  
added). *Creating the appearance* of a single application window differs  
from the claimed subject matter's *actual creation* of a single application  
program configured to provide a single navigable window and multiple  
different functionalities.

1 Accordingly, for at least these reasons, this claim is not anticipated  
2 by Koppolu and is allowable.

3 Claims 2-3 and 5-13 depend from claim 4 and, as such, are  
4 allowable as depending from an allowable base claim.

5 Claim 14 recites a computing system comprising [emphasis added]:

- 6
- **a single application program** configured to provide:
    - 7 ○ a single navigable window;
    - 8 ○ multiple different document-centric functionalities to  
which the single navigable window can be navigated  
9 by a user; and
    - 10 ○ a navigation stack that is configured to enable the user  
to navigate the single navigable window back and  
11 forth between different functionalities.

12 In making out the rejection of claim 14, the Office again argues that  
13 this claim is anticipated by Koppolu. Applicant respectfully but strongly  
14 disagrees and traverses the Office's rejection.

15 Applicant respectfully submits that Koppolu does not disclose a  
16 **single application program** as recited in this claim.

17 Accordingly, for at least this reason, this claim is allowable

18 Claims 15-23 depend from claim 14 and, as such, are allowable as  
19 depending from an allowable base claim. In addition, given the  
20 allowability of the base claim, the rejection of claim 23 over the  
21 combination with Starr is not seen to add anything of significance.

22 Claim 24 recites a computing system comprising [emphasis added]:

- 23
- **a single application program** configured to:
    - 24 ○ display a single navigable window for a user to use in  
navigating between **multiple different functionalities** that  
25 can be provided by the **single application program**; and

- 1           o incorporate different *functionalities* in an extensible  
2           manner so that the user can use the single navigable  
3           window to navigate to the different incorporated  
4           functionalities.

5           In making out the rejection of claim 24, the Office argues that  
6           Koppolu anticipates this claim. Applicant respectfully disagrees and  
7           traverses the rejection. Applicant respectfully submits that Koppolu does  
8           not disclose a *single application program* as recited in this claim.

9           Accordingly, for at least this reason, this claim is allowable.

10          Claims 25-33 depend from claim 24 and, as such, are allowable as  
11          depending from an allowable base claim. In addition, given the  
12          allowability of the base claim, the rejection of claim 33 over the  
13          combination with Starr is not seen to add anything of significance.

14          **Claim 34** recites a computing system comprising [emphasis added]:

- 15          • a *network-accessible single application program*;  
16          • a single navigable window provided by the application  
17          program;  
18          • *multiple different functionalities provided by the*  
19          *application program*, the program being configured so that a  
20          user can navigate the single navigable window and interact  
21          with the different functionalities to accomplish different  
22          tasks; and  
23          • a navigation stack that is configured to enable the user to  
24          navigate the single navigable window back and forth  
25          between different functionalities.

26          In making out the rejection of claim 34, the Office again argues that  
27          this claim is anticipated by Koppolu. Applicant respectfully disagrees and  
28          traverses the Office's rejection. Applicant respectfully submits that

1 Koppolu does not disclose a *single application program* as recited in this  
2 claim.

3 Accordingly, for at least this reason, this claim is allowable.

4 Claims 35-39 depend from claim 34 and, as such, are allowable as  
5 depending from an allowable base claim. In addition, given the  
6 allowability of the base claim, the rejection of claim 39 over the  
7 combination with Starr is not seen to add anything of significance.

8  
9 Claim 40 recites a computing system comprising [emphasis added]:

- 10
- 11 • a software platform comprising software that is  
12 configured to provide a *single application program* that  
13 provides:
    - 14 ○ a single navigable window;
    - 15 ○ capabilities to navigate the single navigable window  
16 to different functionalities that can enable a user to  
17 accomplish different tasks;
    - 18 ○ capabilities to manage navigation activities of the  
19 user;
    - 20 ○ capabilities to provide context-sensitive command  
21 sets and change the command sets as a user's context  
22 changes in accordance with the user's navigation  
23 activities; and
    - 24 ○ capabilities to receive and incorporate into the single  
25 application program individual software components  
that comprise *individual different functionalities*.

21 In making out the rejection of claim 40, the Office argues that the  
22 subject matter of this claim is disclosed by Koppolu. Applicant  
23 respectfully disagrees and traverses the rejection. Applicant respectfully  
24  
25



1 submits that Koppolu does not disclose a *single application program* as  
2 recited in this claim.

3 Accordingly, for at least this reason, this claim is allowable.

4 **Claim 41** recites software code embodied on a computer-readable  
5 medium which, when executed by a computer, provides a user interface  
6 (UI) comprising [emphasis added]:

- 7 • *a single window* that is capable of being navigated to and  
8 between multiple different functionalities that enable a user  
9 to accomplish multiple tasks in connection with *a single*  
10 *application that provides the multiple different*  
11 *functionalities*; and
- 12 • navigation instrumentalities comprising browser-like  
13 navigation buttons that are configured to enable the user to  
14 navigate the single window to and between the multiple  
15 different functionalities.

16 In making out the rejection of claim 41, the Office argues that  
17 Koppolu anticipates this claim. Applicant respectfully disagrees and  
18 traverses the rejection. Applicant respectfully submits that Koppolu does  
19 not disclose a *single application program* as recited in this claim.

20 Accordingly, for at least this reason, this claim is allowable.

21 **Claims 42-45** depend from claim 41 and, as such, are allowable as  
22 depending from an allowable base claim.

23 **Claim 46** recites a computing method comprising [emphasis  
24 added]:

- 25 • displaying a user interface that comprises a *single navigable*  
*window* that can be navigated between *multiple different*  
*functionalities* that are *provided by a single application*  
*program*;

- receiving user input that indicates selection of a particular functionality;
- responsive to receiving said user input, navigating the single navigable window to the particular selected functionality and displaying in said window indicia of said functionality that can enable a user to accomplish a task associated with the particular selected functionality; and
- managing a user's navigation activities using a navigation model that maintains entries that correspond to the user's navigation activities.

In making out the rejection of claim 46, the Office argues that Koppolu anticipates this claim. Applicant respectfully disagrees and traverses the rejection. Applicant respectfully submits that Koppolu does not disclose a *single application program* as recited in this claim.

Accordingly, for at least this reason, this claim is allowable.

Claims 47-63 depend from claim 46 and, as such, are allowable as depending from an allowable base claim.

Claim 64 recites one or more computer-readable media having computer-readable instructions thereon which, when executed by a computer, cause the computer to [emphasis added]:

- display a user interface that comprises:
  - a single navigable window that can be navigated between *multiple different functionalities* that are provided by *a single application program*; and
  - navigation instrumentalities that are configured to enable selection of a particular functionality, the navigation instrumentalities comprising links associated with each of the multiple different functionalities and browser-like navigation buttons that can be used by the user to navigate the single navigable window between the different functionalities;

- receive user input via said navigation instrumentalities that indicates selection of a particular functionality; and
- responsive to receiving said user input, navigate the single navigable window to the particular selected functionality and display in said window indicia of said functionality that can enable a user to accomplish a task associated with the particular selected functionality.

In making out the rejection of claim 64, the Office argues that Koppolu anticipates this claim. Applicant respectfully disagrees and traverses the rejection. Applicant respectfully submits that Koppolu does not disclose a *single application program* as recited in this claim.

Accordingly, for at least this reason, this claim is allowable.

Claims 65-66 depend from claim 64 and, as such, are allowable as depending from an allowable base claim.

Claim 67 recites a computing method comprising [emphasis added]:

- providing a *single application program* that is configured to display a single navigable window *for a user to use in navigating between multiple different functionalities that can be provided by the single application program*; and
- incorporating different functionalities in an extensible manner into the single application program so that the user can use the single navigable window to navigate to the different incorporated functionalities.

In making out the rejection of claim 67, the Office argues that Koppolu anticipates this claim. Applicant respectfully disagrees and traverses the rejection. Applicant respectfully submits that Koppolu does not disclose a *single application program* as recited in this claim.

Accordingly, for at least this reason, this claim is allowable.

1 Claims 68-72 depend from claim 67 and, as such, are allowable as  
2 depending from an allowable base claim.

3 Claim 73 recites a computing method comprising [emphasis  
4 added]:

- 5
- 6 • displaying a user interface that comprises *a single navigable window* that can be navigated between *multiple different document-centric functionalities* that are provided by *a single application program*;
- 7
- 8 • receiving user input that indicates selection of a particular document-centric functionality;
- 9
- 10 • responsive to receiving said user input, navigating the single navigable window to the particular selected document-centric functionality and displaying in said window indicia of said functionality that can enable a user to accomplish a task associated with the particular selected functionality; and
- 11
- 12 • managing a user's navigation activities using a navigation model that maintains entries that correspond to the user's navigation activities.
- 13
- 14

15 In making out the rejection of claim 73, the Office argues that  
16 Koppolu anticipates this claim. Applicant respectfully disagrees and  
17 traverses the rejection. Applicant respectfully submits that Koppolu does  
18 not disclose a *single application program* as recited in this claim.

19 Accordingly, for at least this reason, this claim is allowable.

20 Claims 74-87 from claim 73 and, as such, are allowable as  
21 depending from an allowable base claim.

### 22 Conclusion

23 All of the claims are in condition for allowance. Accordingly,  
24 Applicant requests a Notice of Allowability be issued forthwith. If the  
25

1 Office's next anticipated action is to be anything other than issuance of a  
2 Notice of Allowability, Applicant respectfully requests a telephone call for  
3 the purpose of scheduling an interview.  
4

5 Respectfully submitted,

6  
7 Dated: 6/24/05

8 By: 

Lance R. Sadler  
Reg. No. 38,605  
(509) 324-9256